SI 507 W19 Final Project Sequence

Assignments in Final Project Sequence

The final project sequence has 3 total assignments, as follows:

- **Final Project Proposal** - Due Wednesday, April 10 - total 1000 points
- **Final Project Check-In Assignment** - Due Wednesday, April 17 - total 500 points
- **Final Project (final submission)** - Due Thursday, April 25 - total 4000 points

The following document details all requirements for each. Each submission is on Canvas.

However, in order to do each of the first two assignments, you must, of course, have an understanding of all the Final Project expectations and requirements.

They are detailed here in reverse order from when they are due (first the final project, then the check-in, then the proposal). **Of course, the links above will help you find your way through these instructions.** Please make sure to read them all carefully before asking questions on Piazza!

Overall details

For the final project, you will be planning and executing a Flask application that demonstrates, and allows you to exercise, the skills and tools you have learned and practiced over the course of the semester.

It also provides an opportunity for you to use even more of your own creativity, for you to leverage what you know about integrating ideas and programmatic systems into one project, for you to learn some new things with the frameworks for learning new things that we have practiced. In the process of working on this project, you will also be learning specific new practices and skills during SI 507 class (the class isn't over, it's just mostly in service of working on this project and gaining relevant skills and knowledge!).

There are specific requirements for each assignment listed above that will allow us to grade your project -- all three of these assignments will be graded by instructional staff using a rubric that follows the point allocations listed here.

You will go through a Final Project Proposal, from which you will receive feedback, and a Final Project Check-In, which will have some opportunities for feedback from instructors or classmates built in to the process, before submitting the Final Project. All three of these are assignments for points.

Beyond the **specific requirements**, we will continue to provide some hints, suggestions, and ideas as well as in-class examples and support. None of those are required -- but it's probably a good idea to pay attention to them.

*The instructions begin with the **requirements** for the final project submission, so you are aware of them when you work on the proposal and check-in.*

Final Project Details (Due April 25th)

Your project absolutely must… (900 points)

- ❑ **Be submitted in a GitHub repository** 提交github的repo，而且4.25之后就不能再提交了
  - ❑ You will provide us a link to the repository itself, and should not make commits to the repository after April 25th and before you receive your grade

Flask的app,在本地跑

- ❑ **Include a working Flask application whose code runs locally on a computer**
  - ❑ You cannot gain full credit for code that does not run, or for any functionality that cannot be tested

包括至少一个test suite文件

- ❑ **Include at least 1 test suite file with reasonable tests in it**
  - ❑ *Reasonable* means *non-trivial* -- it does not fulfill a testing requirement to test whether a variable assigned the value 5 has the value 5. However, this does NOT mean you need to include "perfect" or "complete" tests, because we haven't learned enough about testing for that! This is a chance to TRY and work on something new to demonstrate and practice your new skills. *We will be discussing this in class.*

包括一个requirement.txt文件

- ❑ Include a **requirements.txt** file containing all modules that must be installed in order to run the program on someone else's computer

  Even if you are using a different type of environment, it is still possible to create a requirements.txt file -- if you have any trouble with this, you should contact an instructor and reference instruction pages on Canvas.

README.md文件，有模板

- ❑ **Include a clear and readable README.md file that follows our *README template [TBA]* and contains the following information**
  - ❑ Well-organized details about what your project is and does
  - ❑ Specific description of how to run your application
  - ❑ Specific direction of how to use and deal with the application
  - ❑ A list of all routes available in the application and what they can do/how you can access them/how to interact with them
    - ❑ *Note: we'll show examples of this in class, as well*
  - ❑ Clear instructions about how to run the tests for your project
  - ❑ A clear list of all files and/or directories in the repository
  - ❑ A complete list of the project requirements you have chosen and completed
    - ❑ *Note: we'll show examples of this in class, as well*

包括一个例子的数据库文件

- ❑ **Include a sample .sqlite/.db database file**

数据库的模型图

- ❑ **Include a diagram (in any image form you choose) of your database schema**

  The diagram must be clear, readable, understandable by an instructor, and show information about tables/entities, fields in each table, and relationships between tables.

所有的文件都需要

- ❑ **Include EVERY file needed in order to run the project**

跑出来的结果要截图

- ❑ **Include screenshots and/or clear description of what your project should look like when it is working** so we can compare it to when we run your code
  - ❑ You may include this as part of the README.md

    OR you may include this separately (remember that every single file you submit in the repo should be listed in the README.md)

Your Flask application in your project should have (750 points)

三条不同的路径

- ❑ **At least 3 different routes.**

要让什么都不懂的人知道如何用app

- ❑ **View/s a user can see when the application runs that are legible / understandable for someone who has NOT taken this course.**

  For example, this requirement is NOT fulfilled by showing complex Python dictionaries of nested data on the application -- most people can't look at that and understand what it is.

和至少有两个table的数据库交互

- ❑ **Interactions with a database that has at least 2 tables.**
  - ❑ There must be **at least 1 relationship between 2 tables** in the database structure.
  - ❑ The information stored in the database must be viewed or interacted with in some way in the course of using/viewing the application, but how it is viewed, changed, added to, deleted, is completely up to you in every way.

    For example, it is not acceptable to fulfill requirements by creating a database in your Flask app with several models, populating the data with a separate script that is not related to your Flask app, and having 3 routes in the Flask app that have ABSOLUTELY NOTHING to do with any data in your database. We do not think such a situation is likely, of course.

Your final project should include at least 6 of the following things (your choice) - (2250 points total)

Many of these have various levels of difficulty you can attempt. ANY fulfillment of these requirements is valid and reasonable -- the extent of the difficulty you choose is up to you and what you feel ready for and interested in in this limited time frame. You should prioritize making your project reasonable -- the proposal and check-in are intended to help you do so. There are 13 options here, and how challenging you feel each one is may vary from person to person. You may choose any six of these to include -- as long as you can put together and plan a project with them that makes sense!

使用一个我们没用过的模块

❏ Use of a module we did not use for an assignment or in-class exercise in the course of the SI 507 semester that does something new and allows YOU to learn and explore something new. (Modules built into Python do not count for this -- it must be a module you install using **pip** or similar.)

❏ Use of a new module we did not use for an assignment or in-class exercise in the course of the SI 507 semester that does something new and allows YOU to learn and explore something new. (See above)

If you use two such modules, this can count for two of the 6 things.

继承object

❏ Object definitions using inheritance.

For example, defining a class AND then a class that usefully inherits from it, which you could use in your Flask application in some way/s -- perhaps similar to the work you did in HW3 and Project 1. (Although you could probably make this stuff even more complicated if necessary/useful, with all the things you know now!)

多对多的数据库关系

❏ A many-to-many relationship in the database structure that is relied upon during interaction with the Flask application.

❏ At least one form in the Flask application that allows a user to interact with the form and send data from one place to another AND/OR save new data in the database AND/OR show data processed in some way for a user to see.

模版

❏ Templating in your Flask application *(there must be more than 1 template that uses Jinja template-specific logic, e.g. loop/s or conditional/s with the* ***{% %}*** *and* ***{{ }}*** *syntaxes for this to be fulfilled)*

JS文件调整app

❏ Inclusion of JavaScript files in the application that affect a view in your Flask application in some way. (OK if it is minor, as long as it works.)

❏ Links in the views of the Flask application page/s that allow a user to navigate the application and view all its routes without typing in the URL bar (after first loading http://localhost:5000/ with the application running).

❏ Relevant use of built-in libraries **itertools** and/or **collections,** and/or reliance on iterators and/or generators in your code.

❏ Sourcing of data using web scraping.

❏ Sourcing of data using web REST API request/s.

❏ Sourcing of data using user input AND/OR relying upon data from a downloaded dataset in .CSV format or .JSON format.

❏ Caching of data you continually retrieve from the internet in some way, using either a file or a database.

-You may use **advanced_expiry_caching.py**
-You may use the **requests_cache** library (https://pypi.org/project/requests-cache/ -- *note that if you decide to do so and do so successfully, this also counts for using a module we did not study in this class*)
-You may choose to use any other system for caching data that you want (that works for your needs)
-Note that caching is not REQUIRED, but if you do not implement any, you need to be very careful to ensure that your code will always run within reason and that you are clear about what must be done by a user / another programmer to ensure things will work OK. If you aren't sourcing data from an internet source, you may not need to cache at all!

n.b. In fulfilling these requirements, we ask that you personally be thoughtful and reasonable about what you are using and learning and what you want to get out of this course experience yourself. We (the instructional team) don't want to spend a lot of time assuring you that something will "count". **We trust you to make your own choices, just as we have trusted you to grade your own projects for much of the semester; we are all accomplished people here with individual varying goals!** As far as "reasonable" goes -- importing a module and not using it for anything relevant in the project isn't "using" it, for example -- the choices you make here should affect and inform how your project works and/or is organized, in some way...maybe in a very small way, maybe in a big way.

We will grade based on seeing completion of each of the 6 things you tell us in your README you have selected, points divided evenly among the six. If you complete fewer than 6, it is not possible to get all of these points. Not necessarily a bad thing -- it's *much* more important that the project runs!!

Final project hints and suggestions

- If you want to build on the movie-related application work you have done in Projects 2 and 3, you **are welcome to do so**. It's OK to re-use your own work! You will still have to add *new* things for fulfillment of requirements -- more on top of your project 3, or a new project that starts from your project 2 schema. You can take the opportunity to make something real fancy, a portfolio piece, or something waiting for a great design later on (since this is not a front-end course -- you can still make it look somewhat nice, but we do not care about that in 507).
    - Make sure to cite anything that someone else helped you extensively to write, in a comment.
- If you want to build from work you did e.g. in Project 4 option 1, making an app using data about national parks in some way, you are also welcome to do that!
- Either of those efforts is a great way to build on work you've already thought about.
- You are, however, certainly welcome to expand your own interests in any way and branch out to something new -- a new dataset, a new idea.
- It is also FINE if you use a new dataset, new idea, new schema, and yet formulate a structure that is very similar to work we've done in class. Many apps take the same form with different details! It is useful to understand a structure well enough that you can adapt it for your own purposes and your own models/data/etc.

- While this project must be a Flask application, you can focus your energy on something relevant to your own interests. Just a few ideas to consider:
    - If you are interested in user experience, perhaps you'll want to rely on user data entered through forms. (Perhaps not, too, UX is varied in many ways)
    - If you are interested in data visualization, perhaps you'll want to explore a data charting or other visualization library in Python and use it in some way in the project, or visualize data using HTML
    - If you are interested in data analysis, maybe it'll be interesting to source some complicated data from a website, API, or existing data source and build a complex database schema, and then query it with sqlalchemy to show some aggregate findings in different ways, using Flask as a vehicle for viewing and understanding the data
    - If you are interested in processing data or program design, focusing on complex object oriented design and using the results in your program, like a much more complicated project 1 ++, for example, might be valuable and fun
    - If you are interested in data curation, perhaps you want to use some existing dataset and think about how to gain insights from it / process it / show pieces of it to users depending upon their input / think about its provenance and decide what data to source from the internet via scraping or an API to add
    - If you are focused on building your programming skills and understanding concepts from this semester, it might be great to focus specifically on planning

and executing this project based off of your movie-related work on Projects 2 and 3 with a creative spin on it of your own, or off of your work on Project 4 option 1 (if applicable) -- how can you edit the code you wrote so it's reliable, and leverage this national park data into a Flask app in a cool and useful way?

- Make sure, as you think about your proposal, your plan, and your work throughout this process, that you're scoping your project carefully. We're sure you can come up with some very ambitious ideas. Thinking about something like 1.5 the size of a project you've already done in this course, like Project 3, is a good idea. You have a few weeks… but you don't have forever. You can always add more later. Start small!, and build more when and if it makes sense. Make sure you *can* start small. If you can't, you have a problem to work on. (That's what the proposal and check-in should help with.)

Final Project Submission (100 points for accurate submission)

You should submit to the Canvas assignment…

- **A link to your project GitHub repository containing all of the above.**
  - The link should take us directly to the GitHub repository itself, NOT to a specific file. Please make sure this works -- that when you click on that link, you see a repo home. Like this one from the RunestoneInteractive account: https://github.com/RunestoneInteractive/RunestoneComponents
  - If you want it to be private, you **must** add our collaboration repository as a collaborator also. **DO NOT FORGET TO DO THIS, we are on a tight turnaround time for grading.**

- **Text summarizing (in less than or equal to 3 sentences) what your project is and does, and how you think it went for you.** This should be like a very mini version of your grade justification explanation for other projects. You should NOT submit a grade for this -- just tell us a little bit about what you've done so we know what to expect.

Check-In Assignment Details (Due April 17th)

The Check-In is an opportunity for a few things:

- A chance for code review and to practice and consider skills for code review during class and lab.

- A deadline for some of the set-up work for your project to ensure that you have a good timeline for working on it and enough time to get the support you need and build the structure you need for this substantial project.
- A chance to ensure your scoping for your project is appropriate for you, and a chance to scale back if you need to.
- A chance to turn in some code so you have a specific set of things to discuss, if necessary, with instructors.
- A chance to check in, in words, about what you have done so far and what you have left to do so instructors can help you to scale back if we think it's a good idea to do so for your success in this time frame.

It is worth 500 points in total.

NOTE: On Tuesday, April 16th, the day before this check-in is due, we'll be working on code reviews to support your final projects during class. You should plan to bring what you have so far for your SI507project_tools.py to class, and/or your SI507project_tests.py, and/or another code file you are working on for your project, and be prepared to review others' code and have your own reviewed by a couple of your classmates. We will be demonstrating how to do constructive code reviews, which are all about supporting someone else and helping give them ideas that may help about how you could structure or add to or change their code in the future. You'll have a chance to talk about this process and support each other in moving forward on the final project.

Your Check-In assignment should include

- **A database schema diagram showing the tables you will have in your database, the fields each table will contain, and the relationship(s) between tables, if any. (100 points)**
    - Much like your work from Project(s) 2 and 3.
    - If you decide to use your schema from Project 2 and/or Project 3, that is OK, but you will probably want to make some thoughtful updates to it to make it usable.

- A link to a GitHub repository containing all of the following…
    - **(100 points) A code file containing functions and/or class definitions that are NOT Flask route functions or Flask models** that you are pretty sure you will use/need for your project: **SI507project_tools.py**
        - This should be stuff like:
            - Functions to help you gather data if applicable
            - Functions to help you cache data if applicable
            - Functions to help you process computation of data
            - Class definitions to help you encapsulate and manage data

        - Whether or not they rely on inheritance
    - **At this point, you should plan be 100% sure you have access to ALL of the data you need for your project.** If you are relying on user input data, that's fine. If you are downloading a data set, you should have already downloaded it and be looking at it in order to have made your schema diagram. If you are using scraping and/or API requests, you should have code in the SI507project_tools.py file that proves you have successfully accessed some form of the data you need. Do not wait on this; this is an expectation of the
- **(100 points) A code file SI507project_tests.py** which includes a test suite with
    - Reasonable tests for the code in SI507project_tools.py
    - Tests for any other functionality you want to include that isn't included yet (the tests don't have to pass! They just have to be written for TDD use)
    - Does not have to be A LOT of tests, just enough to check that functions/class methods/etc will work properly, whether or not they do now
- **(100 points) Milestones and issues in the GitHub repo that you began with earlier**
    - If you haven't chosen to rely on this as a method of organization for yourself in the long run, that is OK, but we want to see where you came from when we look at your check-in, so these must be included in what you turn in.
- **(100 points) A draft of your README.md file, using the README template [TBA]**
    - Does not have to be complete
    - Should be well on your way
    - Should explain or give clear indication what you have left to do

- *Not required, but optional:* If you are further along in your project than these specifications, you may submit a repository containing ALL the code files you have completed, including **SI507project_tools.py** -- and more, if you want / if you have committed more work than that. More is fine! You can include in your README the status of your project and any concerns you have.
- If you have (?!) completed your entire project at this time, you may submit your entire project to the check-in. Please note that you are doing so. This is fine to do for credit (but certainly not expected). You must include at least a draft of the README.

Check-In Assignment submission

You should submit to the Canvas assignment:

- A link to a GitHub repository which contains the above listed things, as described
  - SI507project_tools.py
  - SI507project_tests.py
  - README.md
  - A set of milestones and associated issues

- A database schema diagram (an image file, a link to an image file in your README.md, a link to an image file in the submission)

Proposal Assignment Details (Due April 10th)

Your project proposal should include

- **A link to a GitHub repository containing a set of milestones and issues** that, overall, detail a big picture of what you need to do to get your final project done.
  - Should include at least 4 milestones
  - Should include at least 8 total issues, most of which ought to be associated with specific milestones
  - These specifics are here so we know you've organized your plan and are ready to work on the project, and that you have some confidence it has reasonable scope.
  - **Note** that each of these final project assignments requires you to submit a link to a GitHub repository. I strongly suggest that these are all the same repository -- that you start with this, and add things to it, bit by bit, as you progress through the project.
- A link to your own document copy of **this Dropbox Paper template** with your answers to the project proposal questions in that template.

Project proposal submission

You should submit to the Canvas assignment:

- A link to the GitHub repository, as noted above, with issues and milestones in it (250 points)
- A link to your Dropbox Paper document on Dropbox Paper that is a copy of the project proposal template (750 points for answering all questions thoughtfully -- not graded for correctness, graded for effort/clarity. No need to be lengthy, concise and clear is great.)

We will try to turnaround (BRIEF) feedback as quickly as possible. *You are not guaranteed feedback that is as timely if this is submitted late, but we'll try.*